UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Método de extracción de redes semánticas de Wikipedia para proporcionar recomendaciones sobre dominios cruzados

Autor: Antonio Navarro Fernández-Aceytuno

Tutor: Iván Cantador Gutierrez

Enero 2015

Resumen

La recomendación sobre dominios cruzados consiste en que a partir de ítems (películas, libros, canciones, etc.) en un dominio dado (cine, literatura, música, etc.) cuya preferencia es conocida para una persona, se sugiera al usuario ítems en otros dominios diferentes que puedan estar relacionados con el primero de algún modo.

En sistemas de recomendación clásicos los ítems sugeridos normalmente hacen referencia entidades (ítems) del mismo dominio (por ejemplo: si tenemos una canción, lo normal es que se nos recomiende otra). Sin embargo, en los novedosos modelos de recomendación sobre dominios cruzados se explora la posibilidad de hallar entidades de otros dominios que estén relacionados de una forma no trivial, como puedan ser películas y libros o música y lugares.

Los sistemas de recomendación son explotados en infinidad de sitios Web de gran popularidad (FilmAffinity[1], Amazon[2], YouTube[3], etc.), que tienen unos modelos de recomendación basados principalmente en los gustos de un usuario. En ellos, un usuario no necesariamente debe haber solicitado recomendaciones, pero debido a la sobrecarga de información y datos en la Web, es siempre necesaria una gestión y filtro de la información existente para sólo dar a los usuarios aquella que les sea útil o relevante. A pesar de la gran eficacia de estos sitios, tienen la desventaja de sólo recuperar ítems en un único dominio, limitando el posible descubrimiento de otros ítems que puedan ser de interés aunque no estén tan directamente relacionados.

En este trabajo se propone explotar la base de conocimiento DBpedia[4], una ontología pública con información extraída de los artículos de la Wikipedia[5]. En concreto, el objetivo del trabajo es extraer taxonomías de DBpedia que engloben categorías de ítems de diferentes dominios y que estén relacionadas entre sí. En particular, los dominios explorados son la música, las películas y los libros. Una vez construidas las taxonomías, éstas se usarán para establecer caminos semánticos entre entidades de dominios diferentes.

Finalmente, con los caminos semánticos entre entidades, se construye un prototipo de modelo de recomendación sobre dominios cruzados. Este prototipo consiste en un método de ranking basado en grafo, que pretende puntuar y ordenar las entidades que más se aproximan "semánticamente", para su posterior recomendación. De este modo, el trabajo explora de forma preliminar el problema de enlazar entidades de distintos dominios que no tienen relaciones obvias entre ellos.

Palabras clave

Taxonomía, red semántica, ontología, grafo, ranking, sistemas de recomendación.

---

[1] http://www.filmaffinity.com

[2] http://www.amazon.es

[3] https://www.youtube.com

[4] http://DBpedia.org

[5] http://www.wikipedia.org

Abstract

Cross-domain recommendation considers a user's preferred items (movies, books, songs, etc.) in a particular source domain (movies, literature, music, etc.) for suggesting items in other target domains, which can be related somehow to the source one.

In classical recommender systems, suggested items usually refer to entities (items) in the same domain (i.e., if we have a song, it is usual to be recommended another song). However, the novel models of cross-domain recommendation aims to find and suggest entities from other domains that are related in a non trivial way, such as movies and books or music and places.

Recommender systems are already been exploited in countless popular Web sites (such as FilmAffinity, Amazon, and YouTube), which have recommendation models based on the users' tastes. In these systems, a user does not need to request item recommendations, but due to the overload of information and data on the Web, it is always necessary to manage and filter existing information to present users with only useful or relevant information. Despite the high efficiency of these systems, they have the disadvantage of only retrieving items in a single domain, limiting the possible discovery of other items that may be of interest, even though they are not directly related.

In this paper we propose to exploit the DBpedia knowledge base, a public ontology with information extracted from Wikipedia articles. Specifically, the study aims to extract inter-linked taxonomies from DBpedia with categories of items from different domains. In particular, we explore the music, movies and books domains. Once the taxonomies are built, they will be used to establish semantic paths between entities in different domains.

Finally, the obtained semantic paths between entities are used to build a prototype cross-domain recommendation model. This prototype uses a graph-based ranking method that aims to rate and rank entities that are "semantically" close, for further recommendation. Hence, the work preliminarily explores the problem of linking entities in different domains that have no obvious relationship between them.

Keywords

Taxonomy, semantic network, ontology, graph, ranking, recommender systems.

Contenido

1. Introducción 1
1.1 Motivación 1
1.2 Preguntas a investigar 2
1.3 Visión general de la propuesta 3
1.3.1 Fase 1: construcción de herramienta para hallar taxonomías 3
1.3.2 Fase 2: Relación semántica 3
1.3.3 Fase 3: Clasificación y recomendación 4
1.4 Estructura del documento 5
2. Trabajo relacionado 6
2.1. Sistemas de recomendación 6
2.2 Similitud semántica entre dominio 7
2.2.1 Medidas estadísticas 7
2.2.2 Medidas topológicas 8
3. Solución propuesta 10
3.1. Representación del conocimiento 10
3.1.1 Redes de clases 10
3.1.2 Redes de instancias 11
3.2. Extracción de conocimiento 13
3.3. Cómputo de la similitud semántica 16
3.4. Recomendación de entidades 19
4. Validación de la solución 20
4.1. Conjunto de datos generado 20
4.2. Ejemplos de recomendaciones 22
4.3. Discusión 29
5. Conclusiones 30
5.1 Resumen 30
5.2 Resultados 30
5.3 Trabajo futuro 31
Bibliografia 33
Anexo A: Herramienta de obtención de taxonomías 35

Índice de tablas

Tabla 1: tablas con las categorías raíz de cada dominio. 20
Tabla 2: Ejemplo de patrones permitidos 21
Tabla 3: Ejemplo de patrones prohibidos 21
Tabla 4: Tabla de compresión de las categorías 22
Tabla 5: Ejemplo de fichero de depuración, para evitar relaciones entre términos muy comunes en las taxonomías. 22
Tabla 6: Tabla de clasificación para el libro El gran Gatsby 23
Tabla 7: Tabla de clasificación para el libro Biblia 24
Tabla 8: Tabla de clasificación para el artista Eminem 24
Tabla 9: Tabla de clasificación para el artista The Beatles 24
Tabla 10: Tabla de clasificación para la artista Rihanna. 25
Tabla 11: Tabla de clasificación para el artista Ozzy Ousborne 25
Tabla 12: Tabla de clasificación para la película Regreso al futuro. 27
Tabla 13: Tabla de clasificación para el libro Jane Eyre 27
Tabla 14: Recomendación para la película Pineapple Express 28

Índice de figuras

Figura 1: tipo de relación entre una instancia del dominio origen a otra instancia del dominio destino a través de una relación semántica.....11

Figura 2: contiene los caminos desde una categoría hacia las demás, la categoría inicial es 2000s_fantasy_novels desde esta trazamos caminos a las categorías del segundo dominio y a sus categorías padre, diferenciadas entre puntuadas o rectas. Se puede observar que establecemos enlaces solo en las categorías con mayor parecido. ..12

Figura 3: Parte del archivo del Linked Data repository.....13

Figura 4: guardamos en x todos los resultados de categorías que tengan la propiedad 'skos:broader' que significaría 'es subcategoría de', por tanto hallaríamos todas las subcategorías de Films_based_on_literature. 13

Figura 5: Pseudocodigo 14

Figura 6: Figura que representa una taxonomía 15

Figura 7: Representación de un enlace directo y un enlace indirecto. 16

Figura 8: De una relación hallada en las taxonomías de películas y libros. En ella el coeficiente de relación es 1 ya que el resto de palabras son reservadas y solo relaciona 'Irán'. 17

Figura 9: Ejemplo de cómputo de un camino hallado con su puntuación final.....19

Figura 10: Recomendación para la película Back To the Future → Dirk Gently's holistic detective Agency.....30

Glosario

DBpedia	Es un proyecto para la extracción de datos de Wikipedia para proponer una versión Web semántica.
Java	Lenguaje de programación orientado a objetos
Linked Open Data	Conjunto de bases de conocimiento estructuradas en formato RDF enlazados entre sí, que serán los que utilizaremos a la hora de extraer taxonomías.
RDF	Es una familia de especificaciones de la World Wide Web Consortium (W3C) originalmente diseñado como un modelo de datos para metadatos
SPARQL	Un acrónimo del inglés SPARQL Protocol and RDF Query Language, es el lenguaje de consulta sobre datos en formato RDF, normalizado por el RDF Data Access Working Group (DAWG) del World Wide Web Consortium (W3C).
SQL	Structured Query Language. Lenguaje declarativo que permite manejar las bases de datos relacionales.
Taxonomía	Un conjunto de categorías organizados de forma jerárquica, usado para organizar información y principalmente destinado a la exploración y visualización de información.

1. Introducción

1.1 Motivación

La tarea de los sistemas de recomendación consiste en que a partir de un ítems (películas, libros, canciones, etc.) de interés para un usuario, se llegue y sugiera otros ítems que pueda estar relacionados con los primeros de una alguna manera, cuando en general estos ítem hacen referencia a entidades del mismo dominio; por ejemplo, si tenemos una canción, lo normal es que se nos recomiende otra.

Sin embargo, en los dominios cruzados exploramos la posibilidad de hallar ítems de otros dominios que estén relacionados no tan directamente, como puedan ser películas y libros o música y lugares.

En los últimos años los sistemas de recomendación han sido utilizados en grandes aplicaciones de comercio online y webs de entretenimiento, como pueden ser Amazon, Netflix[6] y YouTube. La mayoría de ellos, sin embargo, sólo ofrecen recomendaciones entre ítems que pertenecen exclusivamente a un dominio único. Por ejemplo, FilmAffinity ofrece a sus usuarios películas que podrían interesarle basándose en como el usuario ha puntuado otras películas del mismo género o tipo.

Esto no es un problema en este tipo de sitios, ya que éstas sólo se enfocan en un único dominio. Pero, por otra parte, existen sitios multi-dominio, donde los ítems mostrados son de diversa índole. Es el caso de Amazon, en el que si sería útil no sólo recomendar ítems directamente relacionados, si no ítems que puedan tener una relación estrecha y no tener por qué pertenecer al mismo dominio. Un ejemplo de esto podría ser que al comprar un CD de música, aparte de mostrar músicos parecidos, ofrecer más variedad mostrando libros que pueden ser apetecibles, por tener cierta relación no trivial con el tupo de música elegido.

Algunos sitios web ya ofrecen recomendaciones de ítems en diferentes dominios, pero en general, para construir una recomendación de un ítem en un dominio particular, explotan la información del propio usuario -es decir, las preferencias, contexto- en ese dominio o tienen en cuenta una situación de varios dominios se aplican técnicas de filtrado colaborativo (CF) (Desrosiers y Karypis, 2011; Herlocker et al,1999) sobre los conjuntos de datos en los que hay ciertas preferencias del usuario (valoraciones) superpuestos entre dominios. Los sistemas no necesitan información sobre los atributos de ítems, y evitan la dificultad de tratar con la heterogeneidad de datos existente.

[6] https://www.netflix.com/

1.2 Preguntas a investigar

El objetivo del trabajo es extraer taxonomías que engloben categorías de ítems de diferentes dominios y que estén relacionadas entre sí. En particular, los dominios explorados son la música, las películas y los libros. Una vez construidas las taxonomías, éstas se usarán para establecer caminos semánticos entre entidades de dominios diferentes.

Para relacionarlos utilizaremos el concepto de relación semántica, en la que el parecido entre un ítem y otro sea la forma más directa de relacionar dos ítems.

Al investigar sobre cómo podemos construir esta red nos surgen varías preguntas a las que intentaremos dar respuesta.

Pregunta 1: ¿Cómo representar de forma genérica relaciones semánticas entre entidades de diferente dominio?

La representación será mediante un grafo construido a partir de taxonomías de uno y otro dominio. Deberemos definir un modelo de representación de datos independiente del dominio de estos para poder trabajar con ellos.

Pregunta 2: ¿Cómo identificar las relaciones entre un ítem de un dominio y otro de un dominio distinto?

Primero deberemos encontrar un nexo común a investigar, en los dominios estudiados. Algún tipo de relación general que tengan los ítems para poder relacionarlos, por ejemplo el tema del que trata. Una vez con esto ya podemos establecer relaciones semánticas y tener la certeza de que los ítems con los que trabajemos tienen una relación. Las relaciones semánticas trabajan con el parecido entre categorías, es decir, si una película es del género romántico, la canción que encontremos es muy probable que sea del género romántico (suponiendo que estamos investigando la relación películas-música).

Pregunta 3: ¿Qué ocurre si la red construida tiene muchos más caminos hacía ciertos nodos que hacia otros?

Esto es más un problema del set de datos con el que estemos trabajando. En el caso de Wikipedia, las categorías se suelen introducir de forma manual siguiendo unas nomenclaturas. Esto hace que haya categorías muy repetidas y otras que solo aparezcan una vez, dificultando que haya caminos numerosos para estas últimas. Esto hace que la red no pueda ser homogéneamente repartida, ya que los datos con los que trabajamos no son uniformes y varían.

Para poder evaluar estas y otras preguntas necesitaremos probar en un contexto específico con dominios definidos la propuesta. En este caso utilizaremos DBpedia, Web semántica desarrollada sobre Wikipedia en la que obtenemos información sobre esta y nos basaremos en su árbol de categorías. Los dominios que investigaremos a lo largo de este documento serán libros, música y películas.

1.3 Visión general de la propuesta

El trabajo se divide en tres fases:

1.3.1 Fase 1: construcción de herramienta para hallar taxonomías

En esta fase nos centramos en la extracción del conocimiento, es decir que datos nos interesan y como extraerlos. En DBpedia hay mucha información de diferente tipo, en este trabajo utilizamos los conceptos de instancia y categoría siendo los que vamos a utilizar en nuestra extracción. Una instancia en este caso será un artículo que no tenga subcategorías (por ejemplo, un libro). Una categoría será el grado de jerarquía que englobe una cierta cantidad de instancias o de categorías. Es decir tratamos con datos jerarquizados, en el que habrá un ítem raíz y muchos ítems hijos formando un árbol multicamino. También utilizaremos las propiedades particulares de DBpedia (podemos realizar consultas por propiedad de subcategorías, categorías padre, o su equivalente en otro idioma, autores, genero, etc…) para extraer la información. El repositorio que utilizamos de DBpedia se compone de tripletas en formato RDF a las que se acceden mediante un lenguaje basado en SQL, SPARQL, este lenguaje admite multitud de consultas y variables utilizando unos prefijos sobre los cuales vamos a realizar las consultas. Se puede utilizar fácilmente, haciendo consultas en la propia DBpedia desde su servicio de consulta online[7].

Una vez comprendido el funcionamiento del repositorio y como hacer consultas en él, escogemos categorías raíz que serán las que relacionen dominios entre ellos, para esto hemos decidido que las categorías raíz que escojamos deben tener una categoría padre común, para así poder relacionarlas con facilidad. A la hora de extraer los datos, realizamos una búsqueda en profundidad desde las categorías raíz, hasta un límite marcado por el usuario. Además para refinar la obtención de categorías dentro de nuestra taxonomía, solo copiamos las que son de interés. Para esto utilizamos unos patrones de permisibilidad que hacen que si una categoría contiene o no una palabra patrón, está será introducida en la taxonomía o no, en función de a qué tipo de patrón pertenezca, en nuestro caso utilizamos dos tipos de patrones, permitido y prohibido. Una vez realizada la extracción de datos y guardado de estos, se pasa a la fase 2.

1.3.2 Fase 2: Relación semántica

Una vez tenemos los datos de dos dominios concretos, debemos relacionarlos entre ellos para poder crear caminos semánticos y poder construir nuestra red. Para este cometido, creamos dos tipos de vínculos entre categorías: un vínculo directo y un vínculo indirecto. Un vínculo directo será aquel en el que las categorías de ambas taxonomías coincidan literalmente, un vínculo indirecto será relacionar categorías que tengan alguna palabra coincidente, obviando siempre artículos, preposiciones y palabras genéricas como pueden ser ‘música’ o ‘album’. Al hallar esos vínculos ya podremos construir el grafo de caminos semánticos entre las dos taxonomías.

[7] http://DBpedia.org/sparql

1.3.3 Fase 3: Clasificación y recomendación

En esta fase hallaremos finalmente una clasificación de ítems relacionados a un ítem concreto y recomendaremos el que más puntuación haya obtenido en el ranking.

Para cada categoría del ítem inicial generaremos un grafo de caminos hacía categorías de la taxonomía del segundo dominio. A la hora de crear el grafo, vamos añadiendo a cada enlace unos atributos para poder medir el grado de relación entre categorías y hacer mejor la estimación de que categoría se asemeja más a una dada. Una vez establecidos los pesos para cada enlace del grafo evaluamos categoría a categoría del ítem inicial los caminos a otras categorías. Al evaluar, utilizamos el método de puntuación que cada vez que encuentre un camino desde la categoría 1 a una cierta categoría 2 perteneciente a los ítems del segundo dominio se le añada una coincidencia encontrada donde aplicamos el algoritmo de Dijkstra[8] para hallar la distancia más corta, cuanto menor sea la distancia mayor será el peso añadido que sumaremos a la clasificación de ese ítem. Finalmente establecemos la clasificación de ítems y el ítem con mayor puntuación será recomendado.

[8] http://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra

1.4 Estructura del documento

El memorándum de este documento está estructurado de la siguiente manera:

- En la sección 2 se proporciona un resumen de los trabajos correspondientes. Específicamente, revisamos anteriores trabajos sobre métricas de relación semántica –diferenciando similitudes estadística y topológica semánticamente-, y describir brevemente los principales tipos de sistemas de recomendación – dando mayor relevancia en los que están más relacionadas con nuestro enfoque, es decir, basada en el conocimiento, el contexto-consciente, y sistemas de recomendación de dominios cruzados.
- En la sección 3 se presenta el marco basado en semántica propuesto para varios dominios de recomendación, una instancia para el caso de estudio de libros que nos sugiere una relación con películas. En cada subsección, se describe un componente particular de nuestro marco. En concreto, se describe la representación del conocimiento utilizado, los métodos de extracción y los algoritmos investigados para computar entre dominios la relación semántica y recomendaciones personalizadas.
- En la sección 4 se presenta el trabajo experimental realizado para validar nuestro marco en el propuesto caso de estudio.
- Por último, en la sección 5 proporcionamos las conclusiones de este trabajo, y algunas orientaciones para futuras investigaciones.

2. Trabajo relacionado

2.1. Sistemas de recomendación

Los Sistemas de Recomendación (SR) son herramientas de software y técnicas que proporcionan sugerencias para que los artículos sean de utilidad para el usuario. Las sugerencias se refieren a diversos procesos de toma de decisiones, como qué artículos para comprar, qué música escuchar, o qué noticias online leer (Recommender Systems HandBook, Springer). Existen cuatro tipos de sistemas de recomendación:

- Basados en contenido: Al usuario se le recomendarán artículos similares a los que ya votó favorablemente en el pasado. (Adomavicius,2005)
- De Filtrado Colaborativo: Al usuario se le recomendarán items que personas con gustos parecidos y preferencias votaron favorablemente en el pasado. (Adomavicius,2005)
- Basados en conocimiento: El recomendador posee una información en orden de recomendar un artículo. En este trabajo desarrollaremos este tipo de recomendador.
- Híbridos: Combinación entre dos o más de los anteriores.

En general, los SR se aplican a un único dominio, esto es, si votamos películas se nos recomendarán películas, sin embargo de reciente interés los recomendadores sobre dominios cruzados lidian con el problema del enfoque unilateral de los anteriores, enriqueciendo la recomendación. Un sistema de recomendación de dominios cruzados es una extensión de los sistemas de recomendación pero relacionando artículos de diferente índole. Estos sistemas ya los utiliza la Web de Amazon a la hora de recomendar compras de diferentes artículos, que guardan una cierta relación.

Por otro lado los principales problemas que atrae la recomendación sobre dominios cruzados son los siguientes (IBM TJ Watson Research Center):

- Escasa conexión: Los dominios cruzados suelen tener pocas conexiones entre ellos, es un problema establecer un punto de unión entre diferentes dominios.
- Experiencia poco complementaria: Los colaboradores entre dominios cruzados suelen tener diferentes conocimientos e intereses.
- Asimetría: Al establecer puntos de unión, descartamos un grupo amplio de artículos que pueden ser recomendables debido a que no coinciden con ese nexo.

Los sistemas de recomendación sobre dominios cruzado pueden solventar en parte el problema del ‘Arranque en frío’ que consiste en que al tener muy poca información desde la que partir para la recomendación en un primer momento, esta no es eficiente. La posibilidad de añadir este extra, puede hacer que el problema se solucione teniendo más dominios en los que buscar aparecerán, por ende, más artículos expuestos a ser recomendados.

En este trabajo, implementamos la red propuesta para construir automáticamente redes semánticas sobre dominios cruzados con información estructurada extraída de repositorios Linked Data. Esta red conecta conceptos de múltiples dominios a través de propiedades semánticas estableciendo así puentes que pueden ser usados para apoyar o efectuar recomendación de dominios cruzados.

2.2 Similitud semántica entre dominio

Similitud semántica o la relación semántica es un concepto por el cual un conjunto de documentos o términos que figuran en listas de términos se les asigna una métrica basada en la semejanza de su significado/contenido semántico. Sirve en definitiva para medir la diferencia o parecido entre una palabra y otra, normalmente medido sobre una escala entre -1 y 1 siendo 1 la similitud total entre dos palabras. Sin embargo estos dos términos tienen sutiles diferencias, similitud semántica abarca solo relaciones directas (por ejemplo: es, subclase de) mientras que relación semántica admite relaciones más arbitrarias siempre que estén dentro de un mismo contexto. A la hora de medir la similitud entre dos ítems hay dos tipos de medidas: estadísticas y topológicas.

2.2.1 Medidas estadísticas

Las medidas estadísticas se basan en medir la relación semántica entre dos conceptos para poder correlacionar palabras y contextos textuales de un corpus de texto. La gran mayoría de las veces se basa en Modelo de Espacio Vectorial (en inglés VSM) que consiste en que un documento d_i se representa como un vector $d_i = (d_{i1}, d_{i2,\ldots,} d_{in})$ donde N es el tamaño del vocabulario, número de términos distintos en un documento. El componente j-ésimo, d_{ij} representa la importancia relativa de ese término en el documento. Se representan documentos y consultas en un espacio vectorial $\mathbb{R}\ \mathcal{V}$, donde $\mathcal{V}$ es el vocabulario que contiene el documento. Para poder hallar una ponderación representativa de los términos de un documento que por un lado cuantifique cuán representativo es cada término en el documento y que distinga entre términos comunes y raros se suele utilizar el esquema tf-idf. El tf-idf trata sobre que la importancia del término y de cómo se distingue de los demás, consistiendo en la siguiente formula:

$$d_{ij} = tf(t,d) \cdot idf(t)$$

donde dt es la puntuación del término, $tf(t, d)$ mide la importancia del término en el documento e $idf\ t$ mide el poder de discriminación del término. Más explícitamente la formula sería:

$$d_{ij} = \frac{frec(d_i,d_t)}{max_k frec(d_i,t_k)} \cdot log\frac{N}{n(t_j)}$$

donde $frec(d_i, d_t)$ es la frecuencia del término en el documento, $max_k\ frec(d_i, t_k)$ es la frecuencia máxima de algún término en el documento, donde $n(t_j)$ es el número de documentos en los que t_j aparece, y N el número de documentos en el corpus.

En general, el número de términos en el vocabulario es mucho mayor que el número de documentos en el corpus y la matriz término-documento -con los vectores de documentos como columnas- es muy dispersa. Debido a esto, se realiza un Análisis Latente Semántico (LSA) (Deerwester et al., 1990) o Análisis Semántico explicito (ESA) (Gabrilovich and Markovitch (2007)). LSA es una técnica de reducción de dimensión que encuentra un bajo grado de aproximación a la matriz término-documento mediante la realización de una descomposición en valores singulares. LSA asume que las palabras que tengan un significado parecido estarán próximas en el texto. ESA va más allá relacionando ese conjunto de palabras con artículos de Wikipedia para una mayor identificación de estos, una palabra se representa como un vector columna en la

matriz tf-idf del corpus del texto y un documento (cadena de palabras) se representa como el centroide de los vectores que representan sus palabras.

Finalmente el mejor desarrollo para esta medida es ESA seguido de un LSA ambos utilizando el esquema tf-idf para los pesos.

2.2.2 Medidas topológicas

Las medidas topológicas miden la relación semántica entre dos conceptos al considerar conexiones explícitas de tales conceptos en una ontología o una red semántica. Estas estructuras semánticas de enlace se utilizan para estimar la relación semántica, por ejemplo contando la longitud de la ruta de conexión entre dos conceptos dados y teniendo en cuenta que los caminos más cortos indican una mayor relación que caminos más largos.

Existen dos tipos de enfoques que calculan similitud topológica entre conceptos ontológicos:

- Basado en nodos: en el que las principales fuentes de datos son los nodos y sus propiedades. Se basan en la noción de contenido de información (IC) para calcular en qué medida dos conceptos tienen en común información.
- Basado en enlaces: que utilizan los enlaces y sus tipos como el origen de los datos;

En las medidas basadas en nodos según Resnik (Resnik, 1995) los nodos establecen una jerarquía, en el que se engloban los diferentes nodos con sus propiedades, estas propiedades o etiquetas, están escalados de la siguiente forma, por ejemplo consideremos un coche, existirán dos nodos, ruedas y motor, hijos de este, sin embargo hay otros nodos como vehículo que también comprenden a estos dos nodos, aquí siempre daremos mayor importancia al nodo que comprenda a otros y este sea el más bajo, es decir esté más cerca de los nodos hoja. Así a la hora de dar importancia a un nodo a partir de otros dos podríamos puntuar vehículo y coche pero coche al ser más bajo que vehículo (pueden existir nodos tren, avión) y dar una información más precisa nos quedaremos con ese, es lo que Resnik (Resnik, 1995) llamó el LCS (Lowest common subsumer). Y su grado de relación se mide bajo la siguiente formula:

$$rel(c1, c2) = IC(LCS) \triangleq -log\, P(LCS)$$

Donde P (LCS) es la probabilidad de encontrar un concepto en el corpus del texto. Intuitivamente, cuanto mayor es la probabilidad, más común es el nodo que comprende a otros y menos información nos aporta. En (Resnik, 1995), comenta que la probabilidad se estima como la frecuencia relativa de un concepto:

$$P(c) = \frac{freq(c)}{M}$$

Donde M es el número total de ítems identificados en una colección.

Se han propuesto otras métricas basadas en la noción de contenido de información de Resnik(Resnik, 1995). Lin (1998) presenta una versión normalizada de la métrica de la Resnik (Resnik, 1995) que tiene en cuenta el IC de los conceptos involucrados c_1 y c_2:

$$rel(c1, c2) = \frac{2IC(LCS)}{IC(c1) + IC\ (c2)}$$

En las medidas basadas en enlaces, estas se estiman a partir de la distancia, número de enlaces, entre dos conceptos estimando así la relatividad semántica.

Uno de los primeros trabajos en esta dirección fue realizado por Rada et al. (1989), que calcular la distancia de la ruta más corta así:

$$rel(c1, c2) = 2D - d(c1, c2)$$

Donde D es el camino más largo de la red y d(c1,c2) es el número de enlaces del camino más corto entre los conceptos c1 y c2.

Otras aproximaciones conocidas son:

$$rel(c1, c2) = -\log \frac{d(c1, c2)}{2D}$$

$$rel(c1, c2) = 2 \frac{depth(LCS)}{d(c1, LCS) + d(c2, LCS) + 2\, depth(LCS)}$$

En este trabajo trabajaremos con un método basado en nodos y enlaces con métricas topológicas que explicaremos más adelante utilizando un grafo dirigido.

.

3. Solución propuesta

3.1. Representación del conocimiento

Nuestra estructura genérica para la recomendación de varios dominios se basa en una representación del conocimiento basada en la ontología, es decir, un grafo / red de entidades semánticas de diferentes dominios interconectados por las propiedades de la ontología.

Las entidades pueden ser instancias o clases (también denominadas categorías). Las instancias pueden ser o el ítem desde el que partimos, o el ítem de destino, en este caso películas, libros o música. Las clases son propiedades que ligan las instancias con otras clases o instancias, en el caso de DBpedia son etiquetas que se le dan a un artículo para poder categorizarlo. Por ejemplo: *The Da Vinci Code* de Dan Brown es una instancia de un libro, este a su vez tiene la etiqueta *American crime novels* que sería una clase, esta clase engloba a muchas otras instancias como puede ser *The big sleep* del autor Raymond Chandler. Los enlaces pueden expresar relaciones jerárquicas, por ejemplo, 'subclase de' y 'ejemplo de', o pueden tener significados arbitrarios. Es en estas relaciones jerárquicas en las que basaremos la construcción de la red semántica, hallaremos, a partir de ciertas categorías raíz, enlaces entre categorías realizando consultas al set de datos de DBpedia. Este es un set de datos jerarquizado donde podremos realizar consultas en función de propiedades que nos interesen, en nuestro caso nos fijamos en la propiedad es subcategoría de, así llegamos desde una clase alta a clases más bajas que abarcaran instancias más específicas.

Con la construcción de esta red nuestro objetivo es poder puntuar caminos desde una instancia origen del dominio 1 a una o varias instancias destino del dominio 2 y recomendar la que más puntuación obtenga de las instancias del dominio 2. Un posible ejemplo de este proceso sería elegir un libro del dominio 1 Libros y trazar caminos hacia un dominio 2 que podría ser películas, hallando las instancias finales (películas) que guarden una cierta relación con el libro en cuestión.

A continuación explicamos cómo definimos las redes de clase semántica a través de dominios, y cómo instanciamos redes de clase tales como las redes de instancia con los datos obtenidos de forma automática desde la ontología DBpedia.

3.1.1 Redes de clases

La primera parte de nuestro proyecto se basa en la definición de una red semántica, las clases se obtendrán a partir de una propiedad similar a "es categoría padre de" (en DBpedia es 'is skos:broader of'). Se podrían establecer relaciones de otros tipos tales como temporales o de localización por ejemplo, pero dado que en nuestro caso no es de interés el trabajar con ese tipo de relaciones, ya que nuestros dominios no tienen coincidencia temporal o de localización determinantes utilizaremos en este caso los llamados caminos categóricos.

Los caminos categóricos se basan en la relación jerárquica en forma de árbol entre dos dominios, en nuestro caso, determinamos que una buena manera de hallar relaciones utilizando además la jerarquía, era hallando categorías que denominamos categorías raíz que usaban el mismo tipo de acercamiento. Estas clases halladas son género, fuente y tema (utilizando las inglesas: genre, source y topic/theme). Con esto, enmarcábamos unas categorías que tenían mucha probabilidad de estar ligadas, ya que por ejemplo una canción del género romántico podía estar ligada a una película de ese mismo género.

Ejemplo de un posible camino trazado, siendo los rectángulos instancias y los círculos clases o categorías:

Novela: The Big Sleep

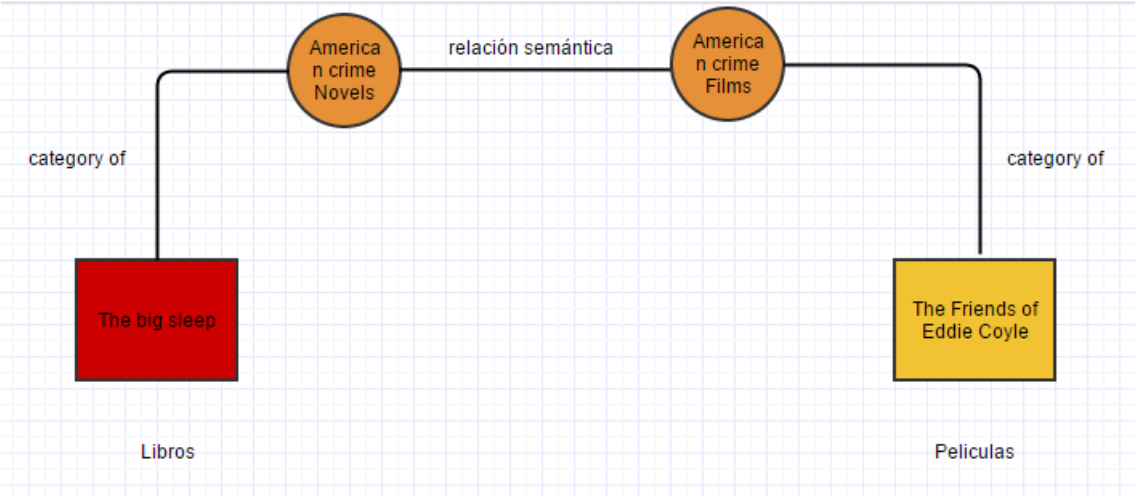

Figura 1: tipo de relación entre una instancia del dominio origen a otra instancia del dominio destino a través de una relación semántica

En nuestro caso asignaremos valores de relevancia a los enlaces, estos valores pueden ser asignados por expertos sobre el dominio en estudio o por usuarios. Aquí primaremos la cercanía de la instancia 1 a la instancia 2 y la especificidad dando más valor a clases directas que a clases padres de estas. A la hora de evaluar que entidades son las más relevantes, utilizamos la propiedad de que aquellas clases que posean caminos más cortos desde la entidad del dominio 1 a otra entidad del dominio 2 serán las que más puntuación obtengan. Es decir si encontramos un camino desde una categoría de la instancia en estudio (por ejemplo: el libro *The big sleep*), a otra categoría de una instancia del dominio 2 sumamos el peso del camino a la instancia 2. Así conforme recorremos los caminos del grafo vamos sumando puntos a las instancias, para finalmente obtener la instancia que más puntuación obtuvo desde la instancia 1.

3.1.2 Redes de instancias

Una vez construidas las redes de categorías, hay que enlazarlas con instancias, para ello leeremos de un fichero posibles instancias del dominio 2 y crearemos una tabla que enlace las instancias del dominio 2 con las categorías del dominio 2. Así enlazamos la red creada entre clases con las instancias del segundo dominio, el primer dominio no es necesario tenerlo enlazado directamente ya que iremos extrayendo instancias una a una evaluando categoría a categoría de esa instancia. El proceso consistirá en por cada categoría de la instancia 1, crearemos un grafo distinto para ver a qué instancias del segundo dominio nos conduce.

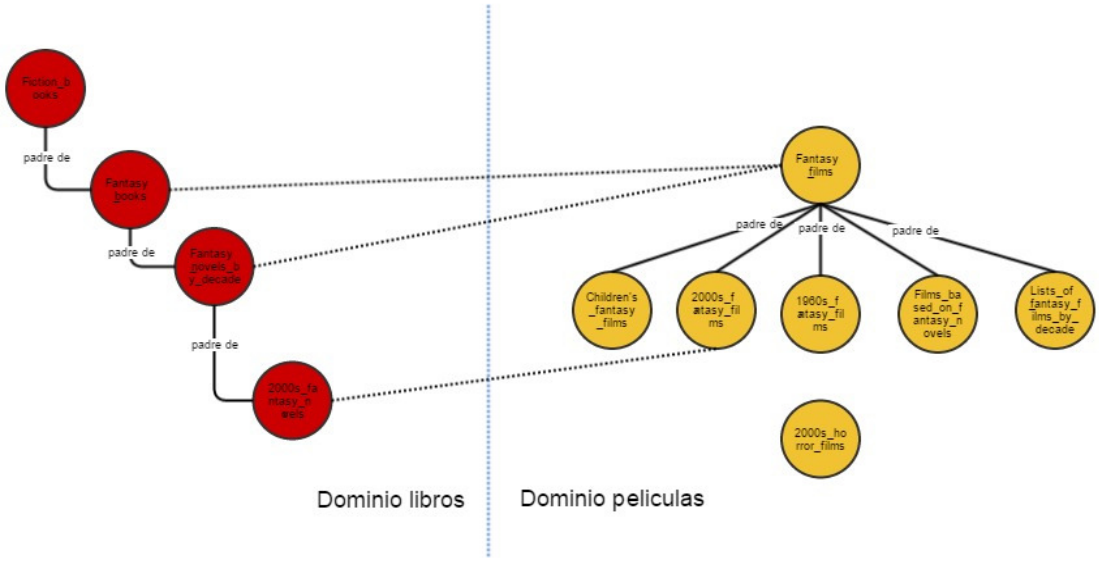

Figura 2: contiene los caminos desde una categoría hacia las demás, la categoría inicial es 2000s_fantasy_novels desde esta trazamos caminos a las categorías del segundo dominio y a sus categorías padre, diferenciadas entre puntuadas o rectas. Se puede observar que establecemos enlaces solo en las categorías con mayor parecido.

Por lo que si tenemos la intención de obtener relaciones fuertes deberemos realizar una comparativa entre categorías a la hora de enlazarlas para únicamente enlazar las más adecuadas y así hacer que el resultado final sea lo más preciso posible.

3.2. Extracción de conocimiento

En esta sección explicamos la forma de obtención y construcción de las taxonomías que utilizamos posteriormente para poder crear nuestra red semántica. La implementación de los métodos propuestos está enfocada a extraer clases de los siguientes tipos: música, películas y libros.

El método de extracción es offline sobre un set de datos en formato RDF, este formato (del inglés *Resource Description Framework*, RDF) es una familia de especificaciones de la *World Wide Web Consortium* (W3C) originalmente diseñado como un modelo de datos para metadatos. El formato es parecido a XML y alberga etiquetas con propiedades de los datos con los que trabajamos:

```
<http://DBpedia.org/resource/Category:Futurama>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2004/02/skos/core#Concept> .
<http://DBpedia.org/resource/Category:Futurama>
<http://www.w3.org/2004/02/skos/core#prefLabel> "Futurama"@en .
<http://DBpedia.org/resource/Category:Futurama>
<http://www.w3.org/2004/02/skos/core#broader>
<http://DBpedia.org/resource/Category:Television_series_created_
by_Matt_Groening> .
<http://DBpedia.org/resource/Category:Futurama>
<http://www.w3.org/2004/02/skos/core#broader>
<http://DBpedia.org/resource/Category:Comic_science_fiction> .
<http://DBpedia.org/resource/Category:Futurama>
<http://www.w3.org/2004/02/skos/core#broader>
http://DBpedia.org/resource/Category:Wikipedia_categories_named_
after_American_animated_television_series
```

Figura 3: Parte del archivo del Linked Data repository.

Para consultar este set de datos utilizamos el lenguaje SPARQL, este lenguaje *basado* en SQL nos permite consultar el archivo de datos fácilmente, un ejemplo de una consulta en SPARQL (***S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage*) podría ser: SPARQL Protocol and RDF Query Language[9]

```
select ?x where { ?x skos:broader <http://dbpedia.org/resource/Category:Films_based_on_literature> }
```

Figura 4: guardamos en x todos los resultados de categorías que tengan la propiedad 'skos:broader' que significaría 'es subcategoría de', por tanto hallaríamos todas las subcategorías de Films_based_on_literature.

En DBpedia un concepto tiene un cierto número de categorías. Estas categorías se establecen manualmente por los administradores de Wikipedia siguiendo una serie de reglas de categorización[10]. Las categorías tienen el problema de que pueden tener errores de etiquetado, o que una categoría se repita en diferentes niveles del árbol de categorías, por lo que tenemos que combatir estos problemas. Para ello, implementamos un algoritmo de búsqueda en profundidad en el que partimos de unas categorías raíz, con un nivel máximo de expansión hacia abajo y unos patrones de permisibilidad para expandir o no una categoría.

[9] http://www.w3.org/TR/rdf-sparql-query

[10] http://en.wikipedia.org/wiki/Wikipedia:FAQ/Categorization

Cuando hablamos de expandir, hablamos de realizar una consulta como la de la figura 4, aquí se puede ver el algoritmo desarrollado:

```
entrada global:
categoriasRaiz: Categorias raiz de la taxonomía
patronesProhibidos: palabras prohibidas para la expansión
patronesPermitidos: palabras aceptadas para la expansión
nivelMaximo: máximo nivel de profundidad o expansión
salida global:
     taxonomias: indice de taxonomias
procedure obtenerTaxonomias:
     taxonomias = new taxonomyIndex();
     foreach raiz en categoriasRaiz
           taxonomias.add(0, null, raiz)
           expandeCategoria(raiz, 0)
procedure expandeCategoria
     entrada local:
           padre: categoria padre a expandir
           nivel: nivel de la categoría padre en la taxonomía
     if nivel < nivelMaximo
           children = query(parent);
     foreach child in children
           if !isForbiddenCategory(child)
                 if isAllowedCategory(child)
                       if !visited(child)
                             taxonomias.add(level, padre, child)
                             expandeCategoria(child, level + 1)
                       else
                             taxonomias.add(level, padre, child)
                 else
                       taxonomias.add(level, padre, child)
                       nietos = query(child)
                       foreach nieto in nietos
                             if isAllowedCategory(nieto)
                                   expandeCategoria(child,level + 1)
```

Figura 5: Pseudocodigo

La razón de realizar una consulta adicional en medio de la expansión es necesaria ya que existen categorías con nombres que darían mucho trabajo introducirlos uno a uno en patrones permitidos como por ejemplo estilos de música: Rock, Pop, Reggae… Por lo tanto consultamos si Rock que no estaría entre los permitidos tiene alguna subcategoría que sí estuviera permitida, en caso de tenerla expandiríamos la categoría Rock, en caso de no tenerla no expandiríamos. Lo ideal sería hacer esta consulta hasta el nivel máximo o tener las palabras dentro del fichero de permitidas, pero lo primero es poco eficiente y lo segundo es un costoso trabajo manual. El grafo es cíclico, en el sentido de que hay casos en los que B es una subcategoría de A, C es una subcategoría de B, y C es una categoría principal de A, lo que implica que A es una subcategoría de B. También intentamos evitar los ciclos en la taxonomía, llevando una cuenta de visitados en la que solo expandimos en el caso de que no hayamos visitado esa categoría anteriormente. Finalmente obtendremos una taxonomía sobre cualquier dominio que le pongamos a la herramienta, estableciendo las categorías raíz obtendremos un índice de categorías que será el que usaremos para nuestro siguiente paso, establecer relaciones semánticas entre categorías.

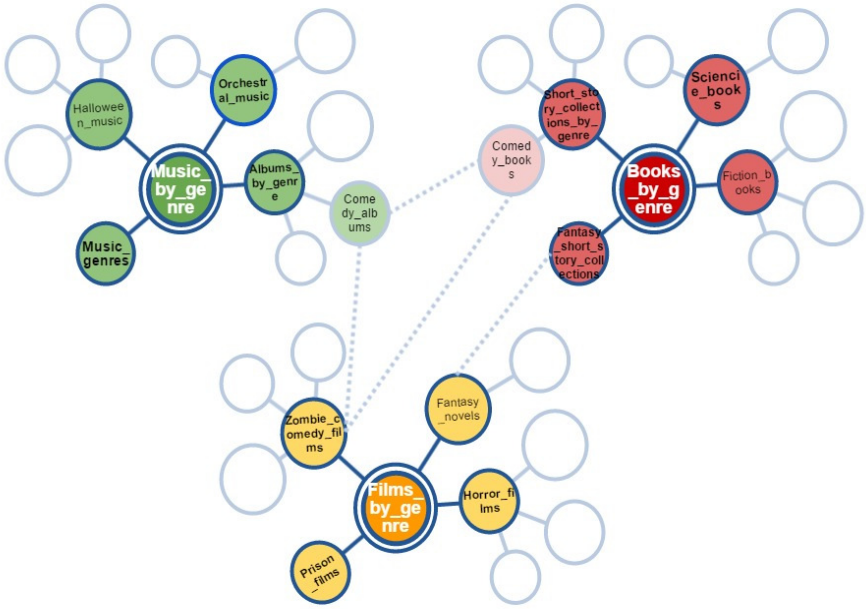

Figura 6: Figura que representa una taxonomía

El principal problema que encontramos en la construcción de las taxonomías, son los ciclos y es que para tener una constancia mayor de que categoría es hija de quien, constatamos que una categoría puede tener varios padres, esto es un problema a la hora de luego relacionar puesto que puede darse que a la hora de crear el grafo debamos recorrer múltiples categorías haciendo que el tiempo de ejecución sea mucho más largo y consigamos relaciones entre categorías que no interesen porque pertenecen a categorías padre muy lejanas. Más adelante trataremos con este problema e intentaremos darle solución, por el momento nuestra decisión fue guardar todas las categorías padre que pueda tener una categoría hija.

Otro gran problema es que las relaciones que establezcamos después puedan ser escasas o repetitivas, en el caso de la relación libros-películas no ocurre tanto este problema ya que la mayoría de las películas existentes se suelen basar en libros, sin embargo, las relaciones música-libros y música-películas son relaciones más problemáticas puesto que no hay una confluencia entre ambos dominios, sin embargo intentamos explotar el hecho de que puedan poseer géneros, temas y fuentes parecidas.

3.3. Cómputo de la similitud semántica

Una vez tenemos las taxonomías de los 3 dominios construidas el siguiente paso es relacionarlas. Para ello creamos una clase (JAVA) genérica en la que estableceremos dos tipos de enlaces entre una taxonomía origen y la taxonomía destino: directos e indirectos.

Los enlaces directos serán aquellos que creemos entre la taxonomía del dominio origen con respecto al destino en el que hallemos una coincidencia completa entre dos categorías de los dominios en estudio. Esto es, si encontramos una categoría A de música que coincide exactamente con una categoría B de películas, entonces obtenemos un enlace directo.

Los enlaces indirectos son un poco más complejos, el mecanismo es parecido a los directos, solo que ahora dividimos las categorías de las taxonomías en palabras, creando así dos diccionarios de palabras, uno por dominio, relacionando las categorías del dominio origen al dominio destino. Para crear los diccionarios deberemos omitir cualquier tipo de palabra que no determine una categoría, es decir palabras comunes en el lenguaje. El lenguaje en el caso de DBpedia es el inglés, por lo que deberemos omitir palabras tales como artículos y preposiciones (*the, of, at*…), además omitiremos todo tipo de palabras que puedan ser de uso habitual en los dominios, por ejemplo '*movie*' podría ser una palabra a evitar ya que aparecerá muy habitualmente en la taxonomía de películas. Esta lista hay que irla refinando conforme se prueba la aplicación para poder ser más precisos a la hora de establecer enlaces indirectos.

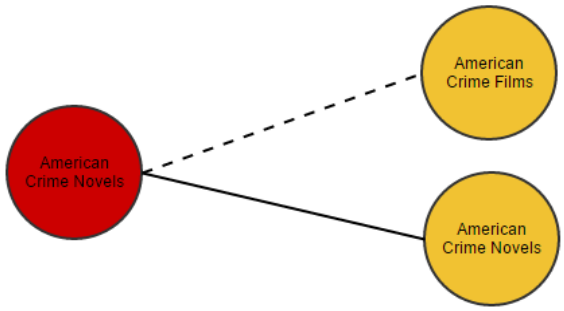

Figura 7: Representación de un enlace directo y un enlace indirecto.

A la hora de relacionar categorías, en un principio sopesamos que todas las categorías que contuvieran una palabra que no estuviera en la lista de palabras comunes (en un fichero de depuración) serían relacionables. Esto hacía que tuviésemos una gran extensión de relaciones, mucho ruido y los grafos generados fuesen muy grandes. Para evitar esto, establecimos que para relacionar una clase con otra necesitábamos que su relación fuera la máxima posible, para ello utilizamos una fórmula de comparación entre vectores, siendo los vectores la lista de palabras de una cierta categoría descartando las mayormente utilizadas.

Un ejemplo de esto podría ser, para la categoría $u \rightarrow (Documentary, films, about, United, States, immigration)$ donde una vez pasado el fichero de depuración nos queda $u \rightarrow (documentary, United, States, Inmigration)$ descartamos 'films' y 'about' por ser muy comunes quedándonos con el resto y se relacionaría con $v \rightarrow (Books, about, United, States, immigration)$ que con el fichero de depuración

nos dejaría $v \rightarrow (United, States, immigration)$, aplicamos la fórmula de comparación entre vectores:

$$\cos(u, v) = \frac{nº\ aciertos}{nº\ maximo\ de\ palabras(u, v)}$$

Dando como resultado 0.75, sí hubiera un caso en el que coincidiesen las palabras clave de ambas categorías dando como resultado 1, esta nueva clase sustituiría a la anterior.

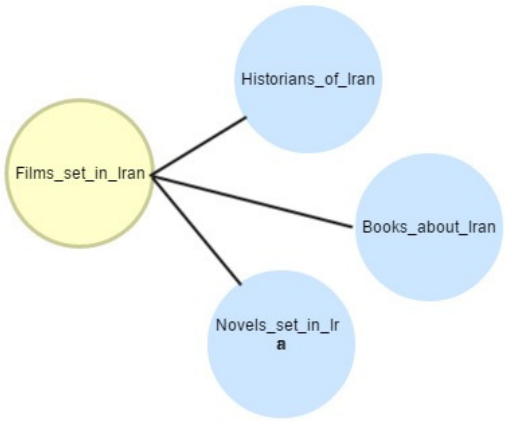

Figura 8: De una relación hallada en las taxonomías de películas y libros. En ella el coeficiente de relación es 1 ya que el resto de palabras son reservadas y solo relaciona 'Irán'.

Cuando ya están todos los enlaces creados, ya podemos crear un grafo dirigido para cada categoría que tenga una instancia del dominio origen. El orden de creación del grafo será el siguiente, 1) Para una categoría dada, la buscamos en la taxonomía de origen, 2) En el caso de pertenecer a la taxonomía, buscaremos si tiene enlaces directos o indirectos 3) Si los tiene, añadimos las categorías al grafo 4) Visitamos las categorías padre de todas las categorías recorridas y seguimos el mismo proceso desde 1) hasta llegar a una categoría límite. Durante la creación del grafo asignamos a cada enlace que creamos un peso, este peso consistirá en dos variables, por un lado el nivel del nodo saliente y por otro el peso del enlace.

A los enlaces, les hemos asignado pesos en función de los siguientes 3 criterios:

- Si el enlace es desde una categoría hija hacia una categoría padre dentro del dominio origen.
- Si el enlace es entre una categoría del dominio origen hacia una categoría del destino.
- Si el enlace es desde una categoría hija hacia una categoría padre dentro del dominio destino.

Además damos más importancia a enlaces directos que a indirectos, dándole más peso (camino más costoso de recorrer) a estos últimos.

Para cada categoría de la instancia origen se creará un grafo (ver figura 2) en busca de caminos hacía categorías destino, las categorías origen y destino las encontramos en dos ficheros de entidades, en los que obtenemos los ítems a recomendar y ser recomendados. Un ítem estará formado por:

- Un nombre de categoría, por ejemplo: *The big sleep*
- Un tipo, ya sea música, libros o películas.
- Una lista de categorías de ese ítem.
- Una puntuación, que será lo que nos sirva más adelante para clasificar los artículos de destino.

El proceso, ahora se resume a ver por cada categoría origen cuantas categorías finales hay, cada vez que exista al menos un camino desde una categoría origen a una destino sumaremos ejecutando el algoritmo de Dijkstra, hallando el camino más corto sobre el grafo entre la categoría origen y la categoría destino en estudio, cuanto más corto sea el camino, con los pesos asignados anteriormente más puntuación extra obtendrá el artículo.

La fórmula de ranking para asignar scores a entidades de destino será:

$$\text{Score}_1(i) = \sum_{i,j}^{N} \frac{1}{distDijsktra(i,j)}$$

$$\text{distDijsktra(i,j)} = \sum_{i,j}^{N} pesoEnlace(i,j)$$

$$\text{pesoEnlace(i,j)} = 0.7 * peso + \left(\frac{MaxLevel}{Level}\right) * 0.3$$

donde 0.7 y 0.3 son estimaciones de la importancia que le damos primero al tipo de enlace y segundo al nivel de expansión donde nos encontramos, cuanto más profundo sea el nivel (más alto) menos peso tendrá por lo tanto será más fácil que al buscar el camino más corto atraviese por ese camino.

3.4. Recomendación de entidades

Esta sección es una leve extensión de la anterior, debemos puntuar para cada artículo del dominio destino en función de la categoría origen, para ello hemos utilizado un algoritmo parecido al de HITS (Kleinberg, 1999), pero sin tener en cuenta el entorno, es decir HITS (Kleinberg, 1999) va calculando nodo a nodo del grafo la suma de multiplicar dos coeficientes en función del entorno para luego quedarse con el más significativo en el nodo destino. En nuestro caso computamos el camino más corto, de menos peso, pero solo asignándole variables que nosotros controlamos como son el nivel, el peso del enlace para finalmente asignar al ítem la distancia del camino más corto desde la categoría de origen a la categoría del ítem destino, explicado con detalle las formulas en la sección anterior.

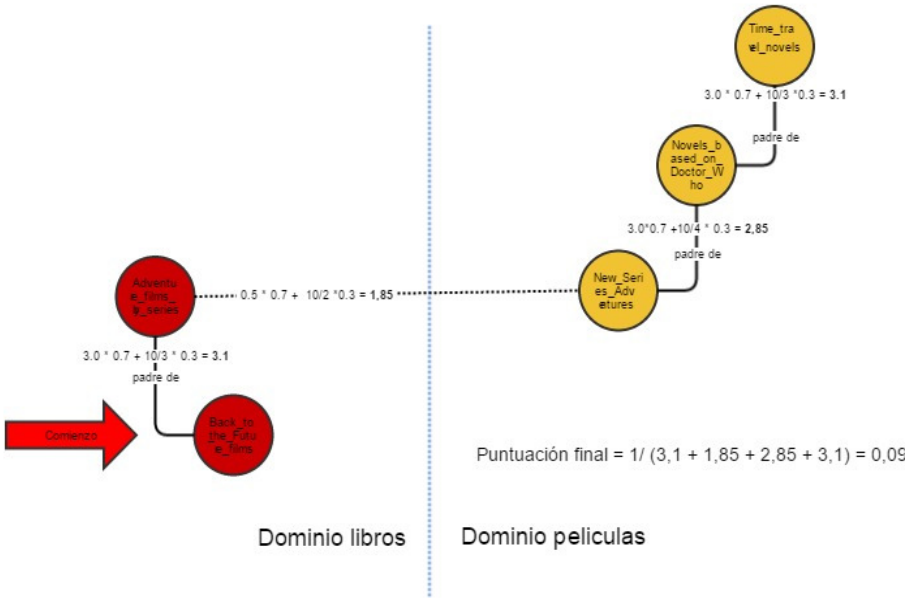

Figura 9: Ejemplo de cómputo de un camino hallado con su puntuación final.

Con las puntuaciones que vamos obteniendo buscamos los ítems que poseen esa categoría entre sus categorías, por lo que vamos consiguiendo una ponderación. Finalmente con las puntuaciones obtenemos la clasificación final para un ítem dado.

4. Validación de la solución

4.1. Conjunto de datos generado

Los 3 dominios de estudio serán: música, libros y películas.

Para cada dominio hayamos categorías raíz con una categoría padre común o con nombre parecido. En este caso hemos elegido como nexo de unión para la posterior creación de la red semántica el género, el tema y la fuente o procedencia. Encontramos varios nexos más pero no decidimos darles relevancia y pueden ser objeto de estudio en otros trabajos. Es el caso de por ejemplo la categoría Fiction_by_topic que englobaría categorías interesantes entre los dominios películas y libros ya que es categoría padre de Topics in films y Topics in literature que podrían ser caminos interesantes a explorar. También otro modo de clasificación entre estos 3 dominios podría ser alguno de los nombrados en la web[11] donde se expone los tipos de clasificaciones que hace Wikipedia sobre sus artículos y categorías.

Esto podrá ser objeto de estudio, puesto que la idea al crear taxonomías es partir desde puntos de uniones parecidos como podrían ser las fechas, el lugar de origen o alguno de nuestros 3 nexos elegidos.

Aquí se pueden ver los resultados obtenidos sobre las categorías que llamaremos raíz:

Música	Tipo
Music_by_genre	Género
Music_by_theme	Tema
Music_by_source	Fuente

Películas	Tipo
Films_by_genre	Género
Films_by_theme	Tema
Films_by_source	Fuente

Libros	Tipo
Books_by_genre	Género
Literature_by_genre	
Books_by_topic	Tema
Literature_by_topic	
Novels_by_source	Fuente

Tabla 1: tablas con las categorías raíz de cada dominio.

Para obtener las taxonomías aplicamos el algoritmo explicado en la figura 5 en ese algoritmo hay dos cosas que debemos tener en consideración: los patrones permitidos y los patrones prohibidos. Estos patrones harán que la taxonomía final de cada categoría quede con el menor ruidos posible, es decir que las categorías que obtengamos tengan relación con el dominio en estudio. Para hacer la depuración más estricta, en lugar de permitir o no permitir palabras también nos fijaremos en la posición de estas, por ejemplo: Si en el dominio películas tenemos una categoría que empieza por 'Books' no nos interesa puesto que lo que buscamos son películas, sin embargo si hay categorías

[11] http://en.wikipedia.org/wiki/Category:Wikipedia_categorization

del tipo Films_based_on_books, que contiene la palabra 'books', aquí si nos interesa puesto que son películas, que es nuestro objetivo. Por tanto hemos realizado una delimitación de la siguiente manera, por un lado los patrones permitidos y prohibidos y dentro de estos 3 categorías: "Empieza por", "contiene" y "Acaba por".

Aquí algunos ejemplos de patrones permitidos y prohibidos:

Permitidos Música	Permitidos Películas	Permitidos Libros	Tipo de patrón
Music	Films	Historians	Empieza por
Songs	film	Myth	Empieza por
albums	about	tales	Contiene
EP	actor	story	Contiene
pianists	movie	collections	Acaba por
ballads	documentaries	ists	Acaba por

Tabla 2: Ejemplo de patrones permitidos

Prohibidos Música	Prohibidos Películas	Prohibidos Libros	Tipo de patrón
People	Books	Decorative	Empieza por
Films	Filmed	Lexicographers	Empieza por
academy	templates	musicians	Contiene
associations	soundtracks	databases	Contiene
record_labels	characters	songwriters	Acaba por
winners	articles	producers	Acaba por

Tabla 3: Ejemplo de patrones prohibidos

Los patrones cambian según el dominio, por ejemplo en películas apenas tuvimos que poner patrones permitidos ya que con *movie* y *film* ya conseguíamos casi todas las categorías relevantes para la taxonomía, en el caso de música tenemos alrededor de unas 180 palabras delimitando lo permitido y lo no permitido, mientras que en el caso de libros son 160. Películas al ser tan delimitable debido a su poca expansión (solo existe desde el siglo XX) nos hace que tengamos un fichero de únicamente 40 palabras.

Una vez tenemos nuestros ficheros de configuración, que incluyen las categorías raíz y los diferentes patrones podemos generar la taxonomía del dominio deseado. Para esto, en la herramienta pondremos un nivel de expansión tan alto como queramos, cuanto mayor es el nivel más clases abarcará la taxonomía, el nivel será un ítem numérico que marcará un límite de expansión, ya que nuestro objetivo es no quedarnos ni muy largos, ni muy cortos, abarcando todas las categorías posibles pero sin necesitar una cantidad ingente de llamadas a la función que harían que el programa no fuera todo lo eficiente que quisiéramos. Finalmente decidimos quedarnos con el nivel 10 de expansión porque consideramos que ya obteníamos suficientes resultados para poder capturar posibles clases pertenecientes a instancias. Como dato, primeramente el fichero generado con la taxonomía con y sin patrones permitidos nos volcaba los siguientes datos:

Dominio	Antes (nº líneas)	Después (nº líneas)
Música	~150k categorías	~55k
Películas	~9k categorías	~8k categorías
Libros	~50k categorías	~22k categorías

Tabla 4: Tabla de compresión de las categorías

Como se puede observar, la reducción de categorías es muy grande entre cuando había ficheros de configuración y cuando no los había. Esta reducción lleva implícita la pérdida de categorías que pueden ser interesantes de abordar, es decisión del usuario de la herramienta el añadir o quitar y valorar si los patrones expuestos aquí eliminan todo el ruido posible con la menor perdida de clases relevantes.

4.2. Ejemplos de recomendaciones

Entre los 3 dominios obtenemos recomendaciones, aunque debido a la multitud de categorías que tienen algunas instancias en el dominio destino es más probable que estén relacionadas con ciertas categorías del dominio origen (por ejemplo, en el dominio libros hay una categoría llamada 'Novels adapted into films', si lo relacionásemos semánticamente sin depurarlo, la palabra films coincidiría con muchas categorías del dominio destino películas y este hecho no nos interesa). Esto puede evitarse utilizando un fichero de depuración completo en el que descartemos palabras a la hora de crear nuestra red semántica, como podrían ser nacionalidades o preposiciones, poco útiles si queremos conseguir relaciones semánticas fuertes.

Preposiciones y conjunciones	Nacionalidades	Palabras clave
for	albanian	films
and	american	books
by	argentinean	music
in	brazilian	movies

Tabla 5: Ejemplo de fichero de depuración, para evitar relaciones entre términos muy comunes en las taxonomías.

Una vez hemos establecido las relaciones y creado los grafos obtenemos las siguientes recomendaciones donde el tipo es el dominio al que pertenece la categoría de ese camino:

Dominio Libros-Películas:

Pos. y caminos	Artículo
1	The_Great_Gatsby_(2013_film)
Camino	Adultery_in_novels tipo1 → Adultery_in_films tipo2
Camino	Novels_set_in_New_York_City tipo1 → Books_about_New_York_City tipo1 → Films_set_in_New_York_City tipo2
Camino	Novels_set_in_the_Roaring_Twenties tipo1 → Films_set_in_the_Roaring_Twenties tipo2
2	Thoroughly_Modern_Millie
Camino	Novels_set_in_New_York_City tipo1 → Books_about_New_York_City tipo1→ Films_set_in_New_York_City tipo2
Camino	Novels_set_in_the_Roaring_Twenties tipo1 → Films_set_in_the_Roaring_Twenties tipo2→ Films_set_in_the_1920s tipo2
Camino	Novels_set_in_the_Roaring_Twenties tipo1 → Films_set_in_the_Roaring_Twenties tipo2
3	New_York_(film)
Camino	Novels_set_in_New_York_City tipo1 → Books_about_New_York_City tipo1→ Films_set_in_New_York_City tipo2
Camino	Novels_set_in_New_York_Citytipo1→Indian_films_set_in_New_York_City tipo2

Tabla 6: Tabla de clasificación para el libro El gran Gatsby

Pos. y caminos	Artículo
1	The_Passion_of_the_Christ
Camino	Bible tipo1 → Christian_texts tipo1 → Religious_texts tipo1 → Religious_epic_films tipo2 → Epic_films tipo2
Camino	Bible tipo1 → Christian_texts tipo1 → Christian_literature tipo1, → Christian_film_festivals tipo2 → Films_about_Christianity tipo2
Camino	Bible tipo1 → Christian_texts tipo1 → Religious_texts tipo1 → Religious_epic_films tipo2
Camino	Bible tipo1 → Films_based_on_the_Bible tipo2 → Films_set_in_Israel tipo2
2	Kingdom_of_Heaven_(film)
Camino	Bible tipo1 → Christian_texts tipo1 → Religious_texts tipo1, → Religious_epic_films tipo2 → Epic_films tipo2
Camino	Bible tipo1 → Christian_texts tipo1 → Religious_texts tipo1, → Religious_comedy_films tipo2 → Films_about_religion tipo2
Camino	Bible tipo1 → Films_based_on_the_Bible tipo2 → Films_set_in_Israel tipo2

3	The_Omen
Camino	Bible tipo1 → Christian_texts tipo1 → Religious_texts tipo1 → Religious_horror_films tipo2
Camino	Bible tipo1 → Films_based_on_the_Bible tipo2 → Films_set_in_Israel tipo2

Tabla 7: Tabla de clasificación para el libro Biblia

- Dominio Música-Libros:

Pos. y caminos	Artículo
1	Pride_and_Prejudice_and_Zombies
Camino	American_comedy_musicians tipo1 → American_comedy_novels tipo2
2	John_Dies_at_the_End
Camino	American_comedy_musicians tipo1 → American_comedy_novels tipo2
3	A_Confederacy_of_Dunces
Camino	American_comedy_musicians tipo1 → American_comedy_novels tipo2

Tabla 8: Tabla de clasificación para el artista Eminem

Pos. y caminos	Artículo
1	On_the_Road
Camino	Beat_groups tipo1 → Beat_novels tipo2
2	Naked_Lunch
Camino	Beat_groups tipo1 → Beat_novels tipo2
3	Junkie_(novel)
Camino	Beat_groups tipo1 → Beat_novels tipo2

Tabla 9: Tabla de clasificación para el artista The Beatles

Música-Películas:

Pos. y caminos	Artículo
1	Stomp_the_Yard
Camino	Dance_musicians tipo1 → American_dance_films tipo2
Camino	Female_hip_hop_musicians tipo1 → Hip_hop_musicians tipo1 → Hip_hop_films tipo2→ African-American_films tipo2
Camino	Female_hip_hop_musicians tipo1 → Hip_hop_films tipo2
Camino	Hip_hop_singers tipo1 → Hip_hop_films tipo2, → African-American_films tipo2
Camino	Hip_hop_singers tipo1 → Hip_hop_films tipo2

2	Save_the_Last_Dance
Camino	Dance_musicians tipo1 → American_dance_films tipo2
Camino	Female_hip_hop_musicians tipo1 → Hip_hop_musicians tipo1→ Hip_hop_films tipo2→ African-American_films tipo2,→ Films_about_race_and_ethnicity tipo2
Camino	Female_hip_hop_musicians tipo1 → Hip_hop_films tipo2
Camino	Hip_hop_singers tipo1 → Hip_hop_films → African-American_films tipo2→ Films_about_race_and_ethnicity tipo2
Camino	Hip_hop_singers tipo1 → Hip_hop_films tipo2
3	Bamboozled
Camino	Female_hip_hop_musicians tipo1 → Hip_hop_musicians tipo1 → Hip_hop_films tipo2→ African-American_films tipo2
Camino	Female_hip_hop_musicians tipo1 → Hip_hop_musicians tipo1, → Hip_hop_films tipo2 → African-American_films tipo2→ Films_about_race_and_ethnicity tipo2
Camino	Female_hip_hop_musicians tipo1 → Hip_hop_films tipo2
Camino	Hip_hop_singers tipo1 → Hip_hop_films tipo2 → African-American_films tipo2
Camino	Hip_hop_singers tipo1 → Hip_hop_films tipo2 → African-American_films tipo2 → Films_about_race_and_ethnicity tipo2
Camino	Hip_hop_singers tipo1 → Hip_hop_films tipo2

Tabla 10: Tabla de clasificación para la artista Rihanna.

Pos. y caminos	Artículo
1	This_Is_Spinal_Tap
Camino	English_heavy_metal_singers tipo1 → English_heavy_metal_musicians tipo1 → Heavy_metal_musicians_by_nationality → Heavy_metal_films tipo2
2	Rock_Star_(2001_film)
Camino	Igual que en pos 1.
3	Airheads
Camino	Igual que en pos 1.

Tabla 11: Tabla de clasificación para el artista Ozzy Ousborne

Películas-Libros:

Pos. y caminos	Artículo
1	Dirk_Gently's_Holistic_Detective_Agency
Camino	Back_to_the_Future_films tipo1 → Comedy_films_by_series tipo1→ Japanese_comedy_novels tipo2→ Comedy_novels tipo2
Camino	Back_to_the_Future_films tipo1 → Time_travel_films tipo1→ Science_fiction_films_by_genre tipo1→ British_science_fiction_novels tipo2
Camino	Back_to_the_Future_films tipo1 → Adventure_films_by_series tipo1→New_Series_Adventures tipo2 → Novels_based_on_Doctor_Who → Time_travel_novels tipo2
Camino	1980s_comedy_films tipo1 → Comedy_films_by_decade tipo1→Comedy_films tipo1 → British_comedy_novels tipo2→ Comedy_novels tipo2
Camino	1980s_comedy_films tipo1 → 1980s_fantasy_novels tipo2
Camino	1980s_comedy_films tipo1 → 1980s_science_fiction_novels tipo2
Camino	1980s_science_fiction_films tipo1 → Science_fiction_films_by_decade → Science_fiction_films tipo1→ Comic_science_fiction_novels tipo2→ Comedy_novels tipo2
Camino	1980s_science_fiction_films tipo1 → 1980s_science_fiction_novels tipo2
Camino	1980s_science_fiction_films tipo1 → Science_fiction_films_by_decade tipo1, → Science_fiction_films tipo1 → British_science_fiction_novels tipo2
Camino	American_teen_comedy_films tipo1 → American_comedy_films → British_comedy_novels tipo2→Comedy_novels tipo2
Camino	Time_travel_films tipo1 → Science_fiction_films_by_genre tipo1→ Comic_science_fiction_novels tipo2 → Comedy_novels tipo2
Camino	Time_travel_films tipo1 → Science_fiction_films_by_genre tipo1→ Science_fiction_films tipo1→Scottish_science_fiction_novels → British_science_fiction_novels tipo2
Camino	Time_travel_films tipo1 → Time_travel_novels tipo2
2	Job:_A_Comedy_of_Justice
Camino	Back_to_the_Future_films tipo1 → American_science_fiction_films tipo1, → Comic_science_fiction_novels tipo2
Camino	Back_to_the_Future_films tipo1 → Comedy_films_by_series tipo1→ The_Emberverse_series tipo2→ American_post-apocalyptic_novels tipo2→ American_science_fiction_novels tipo2
Camino	1980s_comedy_films tipo1 → 1980s_fantasy_novels tipo2
Camino	1980s_comedy_films tipo1→1980s_science_fiction_novels tipo2

Camino	1980s_science_fiction_films tipo1 → Science_fiction_films_by_decade tipo1 → Comic_science_fiction_novels tipo2
Camino	1980s_science_fiction_films tipo1 → 1980s_science_fiction_novels tipo2
Camino	1980s_science_fiction_films tipo1 → Science_fiction_films_by_decade tipo1→ American_science_fiction_novels tipo2
Camino	Time_travel_films tipo1 → Science_fiction_films_by_genre tipo1, → Science_fiction_films tipo1→Comic_science_fiction_novels tipo2
Camino	Time_travel_films tipo1 → Science_fiction_films_by_genre tipo1→American_science_fiction_novels tipo2
3	Many_Waters
Camino	Back_to_the_Future_films tipo1 → Comedy_films_by_series tipo1→Twilight_series tipo2→American_young_adult_novels tipo2
Camino	Back_to_the_Future_films tipo1 → Adventure_films_by_series tipo1 → New_Series_Adventures tipo2 → Novels_based_on_Doctor_Who→y Time_travel_novels tipo2
Camino	1980s_comedy_films tipo1 → 1980s_fantasy_novels tipo2
Camino	1980s_comedy_films tipo1 → 1980s_science_fiction_novels tipo2
Camino	1980s_science_fiction_films tipo1 → 1980s_science_fiction_novels tipo2
Camino	Time_travel_films tipo1 → Time_travel_novels tipo2

Tabla 12: Tabla de clasificación para la película Regreso al futuro.

Libros-Música:

Pos. y caminos	Artículo
1	Alien_Sex_Fiend
Camino	Gothic_novels tipo1 → British_gothic_rock_groups tipo2
Camino	Gothic_novels tipo1 → Gothic_fiction tipo1→ British_gothic_rock_groups tipo2→ British_rock_music_groups_by_style tipo2→ British_rock_music_groups tipo2
2	The_Cure
Camino	Gothic_novels tipo1 →British_gothic_rock_groups tipo2
3	Siouxsie_and_the_Banshees
Camino	Gothic_novels tipo1 → British_gothic_rock_groups tipo2

Tabla 13: Tabla de clasificación para el libro Jane Eyre

Películas-Música:

Pos. y caminos	Artículo
1	Tenacious_D
Camino	2000s_comedy_films tipo1 → American_comedy_musical_groups tipo2
Camino	2000s_comedy_films tipo1 → Comedy_rock tipo2
Camino	American_action_comedy_films tipo1 → Action_comedy_films tipo1 → Comedy_rock tipo2
Camino	American_black_comedy_films tipo1 → American_comedy_musical_groups tipo2
Camino	American_black_comedy_films tipo1 → Black_comedy_films tipo1→Comedy_rock tipo2
Camino	Films_set_in_California tipo1 → Films_set_in_the_United_States_by_state tipo1→ Films_set_in_California tipo1→ Rock_music_groups_from_California tipo2
2	Bloodhound_Gang
Camino	2000s_comedy_films tipo1 → American_comedy_musical_groups tipo2
Camino	2000s_comedy_films tipo1 → Comedy_rock tipo2
Camino	American_action_comedy_films tipo1 → Action_comedy_films tipo1→Comedy_rock tipo2
Camino	American_black_comedy_films tipo1 → American_comedy_musical_groups tipo2
Camino	American_black_comedy_films tipo1 → Black_comedy_films tipo1, Black_comedy_films tipo1 → Comedy_rock tipo2
3	The_Axis_of_Awesome,
Camino	2000s_comedy_films tipo1 → Comedy_rock tipo2
Camino	2000s_comedy_films tipo1 → Australian_comedy_musical_groups tipo2
Camino	American_action_comedy_films tipo1 → Action_comedy_films tipo1→Comedy_rock tipo2
Camino	American_black_comedy_films tipo1 → Black_comedy_films tipo1→Comedy_rock tipo2
Camino	American_black_comedy_films tipo1→Australian_comedy_musical_groups tipo2

Tabla 14: Recomendación para la película Pineapple Express

4.3. Discusión

En esta sección detallamos los principales problemas o limitaciones que tuvimos con los resultados obtenidos.

Los resultados como se puede observar son medianamente satisfactorios, en el caso de la relación libros-películas no hay problema, son dos dominios que tienen una relación muy estrecha, ya que muchas películas están basadas en novelas al igual que comparten muchas categorías en común. Por otro lado la relación entre música con alguno de los otros dos dominios es mucho menor, tienen muchas menores coincidencias, se hallan muchos menos caminos y estos caminos son con ruido muchos de ellos. El constante ruido que obtenemos desde música viene desde la creación de las taxonomías, en música obtuvimos una taxonomía muy grande con muchos términos y al realizar los descartes aunque fue exhaustivo, faltó previsión para ver que categorías podían dar problemas que luego intentamos subsanar con el fichero de depuración, aunque obtenemos algunas relaciones satisfactorias no podemos decir que al relacionar música con otro dominio el resultado vaya a ser siempre el esperado, dando lugar a relaciones escasas como dijimos antes. Además nos encontramos con que hay categorías que tienen una relación total con el dominio destino, estas categorías es necesario delimitarlas ya que por ejemplo si poseen una palabra típica de otro dominio pueden dar lugar a muchas relaciones indeseadas. Por lo tanto dentro del cómputo global los resultados no son todo lo bueno que esperábamos pero si nos atenemos a obtención de taxonomías, generación de redes semánticas y por último recomendación, el resultado final es aceptable.

5. Conclusiones

5.1 Resumen

La recomendación sobre dominios cruzados consiste en que a partir de ítems (películas, libros, canciones, etc.) en un dominio dado (cine, literatura, música, etc.) cuya preferencia es conocida para una persona, se sugiera al usuario ítems en otros dominios diferentes que puedan estar relacionados con el primero de algún modo

En este trabajo, de forma preliminar, hemos desarrollado una herramienta para la obtención de taxonomías desde el Linked Data Repository de DBpedia, esta herramienta nos permite elegir el nivel de expansión y configurar los patrones por los que se regirá la taxonomía, una vez obtenidas las taxonomías de los diferentes dominios, obtuvimos una red semántica de las clases de cada taxonomía relacionándolas entre sí en función de unos parámetros que nosotros mismos introdujimos para poder más adelante puntuar los caminos hallados en cada categoría.

Una vez construida la red semántica de clases, construimos una red de instancias a partir de este en la que relacionamos un ítem de un dominio origen con ítems de un dominio destino. Finalmente puntuamos los ítems del dominio destino para obtener clasificaciones y recomendaciones para el ítem origen.

5.2 Resultados

Conseguimos hallar recomendaciones entre dominios cruzados, una vez halladas las taxonomías y los enlaces entre estas. Los resultados obtenidos son desiguales dependiendo de los dominios en estudio, puesto que como ya se expuso hay dominios con relaciones más dirigidas que en otros. El estudio pretende comprobar la genericidad de estudio de cualquier dominio y comprobamos que siempre que haya categorías raíz similares y posibles relaciones es posible realizar este estudio siempre que los dominios estudiados tengan algo que ver.

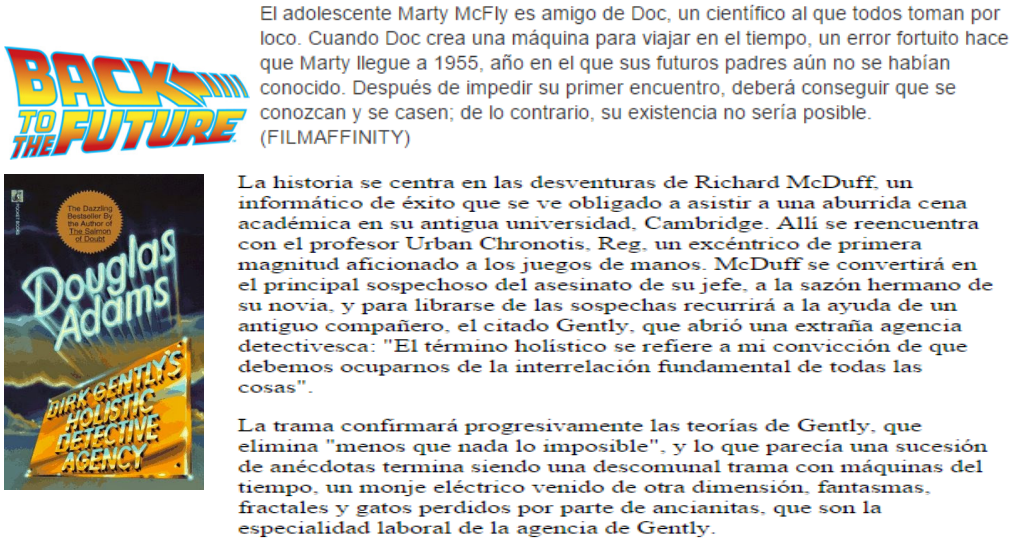

Figura 10: Recomendación para la película Back To the Future → Dirk Gently's holistic detective Agency

5.3 Trabajo futuro

Dentro de un posible marco nuevo, sería necesario investigar todo el concepto de Categorización de Wikipedia, así se podrían establecer taxonomías puente entre dominios que harían más fácil el desarrollo de redes semánticas, es decir en los casos en los que las relaciones no fueran tan estrechas como en películas y libros encontrar un dominio intermedio que ligue una categoría origen de una destino. Además hemos encontrado categorías raíz que serían interesantes de investigar como pueden ser: 'Categories_by_parameter', 'Fiction_by_topic', Topics_in_films y Topics_in_literature.

También el desarrollo de nuevas fórmulas de cómputo para los enlaces y nodos puede hacer que mejoren los resultados.

Añadir que podría ser de interés mezclar este tipo de recomendador con otros con filtrado colaborativo o centrados en el usuario con el fin de que al mezclar ambos el ranking de resultados sea mucho más refinado.

Bibliografia

Desrosiers, C., & Karypis, G. (2011). **A Comprehensive Survey of Neighborhood-based Recommendation Methods.** In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.),

Francesco Ricci, Lior Rokach, Bracha Shapira ·Paul B. Kantor (2011) **Recommender Systems Handbook**. Springer.

Adomavicius, G., & Tuzhilin, A. (2005). **Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions**. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734–749.

Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. **Cross-domain Collaboration Recommendation**

Gabrilovich, E., & Markovitch, S. (2007). **Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis.** In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 1606–1611.

Resnik, P. (1995). **Using Information Content to Evaluate Semantic Similarity in a Taxonomy.** In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995), pp. 448–453.

Lin, D. (1998**). An Information-Theoretic Definition of Similarity.** In Proceedings of the 15th International Conference on Machine Learning (ICML 1998), pp. 296–304.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). **Development and Application of a Metric on Semantic Nets**. IEEE Transactions on Systems, Man, and Cybernetics 19(1), 17–30.

Kleinberg, J. M. (1999**). Hubs, Authorities, and Communities**. ACM Computing Surveys, 31(4es), 5.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). **Indexing by Latent Semantic Analysis**. Journal of the American Society for Information Science, 41(6), 391–407.

Cantador, I., Castells, P., & Bellogín, A. (2011). **An Enhanced Semantic Layer for Hybrid Recommender Systems: Application to News Recommendation**. International Journal on Semantic Web and Information Systems, 7(1), 44–78.

Marius Kaminskas, Ignacio Fernández-Tobías, Francesco Ricci, Iván Cantador. **Ontologybased Identification of Music for Places**. In: Proceedings of the 13th International Conference on Information and Communication Technologies in Tourism (ENTER 2013), pp. 436-447. Springer- Verlag. ISBN 978-3-642-36308-5.

Ignacio Fernández-Tobías, Marius Kaminskas, Iván Cantador, Francesco Ricci. 2011**. A Generic Semantic-based Framework for Cross-domain Recommendation**. In: Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011) ACM Press, pp. 25-32.

Anexo A: Herramienta de obtención de taxonomías